

Enabling All-Node-Repair in Minimum Storage Regenerating Codes

Jie Li, *Student Member, IEEE*, Xiaohu Tang, *Member, IEEE*, and Chao Tian, *Senior Member, IEEE*

Abstract

We consider the problem of constructing exact-repair minimum storage regenerating (MSR) codes, for which both the systematic nodes and parity nodes can be repaired optimally. Although there exist several recent explicit high-rate MSR code constructions (usually with certain restrictions on the coding parameters), quite a few constructions in the literature only allow the optimal repair of systematic nodes. This phenomenon suggests that there might be a barrier between explicitly constructing codes that can only optimally repair systematic nodes and those that can optimally repair both systematic nodes and parity nodes. In the work, we show that this barrier can be completely circumvented by providing a generic transformation that is able to convert any non-binary linear maximum distance separable (MDS) storage codes that can optimally repair only systematic nodes into new MSR codes that can optimally repair all nodes. This transformation does not increase the alphabet size of the original codes, and only increases the sub-packetization by a factor that is equal to the number of parity nodes. Furthermore, the resultant MSR codes also have the optimal access property for all nodes if the original MDS storage codes have the optimal access property for systematic nodes.

Index Terms

Distributed storage, high-rate, MDS codes, MSR codes, optimal access, optimal repair.

I. INTRODUCTION

Distributed storage systems built on a large number of unreliable storage nodes have important applications in large-scale data center settings, such as Facebook's coded Hadoop, Google Colossus, and Microsoft Azure [5], and in peer-to-peer storage settings, such as OceanStore [14], Total Recall [1], and DHash++ [2]. To ensure reliability, redundancy is imperative for these systems. Generally speaking, there are two mechanisms of redundancy, namely replication and erasure coding. Comparing with repetition codes, erasure codes can provide higher reliability at the same redundancy level, and thus are more attractive.

When a storage node fails, a self-sustaining distributed storage system should make a repair to maintain the healthy and continuing operation of the overall system. During the repair process, the *repair bandwidth*, which is defined as the amount of data downloaded from some surviving nodes to repair the failed node, should be minimized. The repair bandwidth of the classic maximum distance separable (MDS) erasure codes, such as Reed-Solomon codes [13], is rather excessive because they only allow a naive repair strategy, i.e., downloading the amount of the original data to first reconstruct the original file, and then to repair the failed node.

Recently, Dimakis *et al.* established a trade-off between the storage and the repair bandwidth in [3]. On the trade-off curve [3], two extremal points are of particular interest, known as the minimum-storage regenerating (MSR) point and the minimum-bandwidth regenerating (MBR) point, the former of which is the focus of this work. A $(k+r, k)$ MSR code with node storage capacity N has the *optimal repair property* that the repair bandwidth is $\gamma = \frac{d}{(d-k+1)}N$, where d ($k \leq d \leq k+r-1$) is the number of helper nodes during the repair process of a failed node. This can be achieved by downloading $\frac{N}{d-k+1}$ symbols from each of any d surviving nodes [3]. Actually, MSR codes can be viewed as MDS storage codes¹ with the optimal repair

J. Li and X.H. Tang are with the Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu, 610031, China (e-mail: jieli873@gmail.com, xhutang@swjtu.edu.cn).

C. Tian is with the Department of Electrical Engineering and Computer Science, The University of Tennessee-Knoxville, Knoxville, TN 37996, USA (e-mail: chao.tian@utk.edu).

¹In this paper, MDS storage codes considered are all vector codes.

property. Indeed, the codes proposed in [3] are functional-repair codes, which allow the repaired node to have different but functionally-equivalent content from the failed one; however, exact-repair MSR codes, which require the repaired node to have exactly the same content as the failed node, can simplify system designs in practice, and thus are preferred, see [9], [10], [15], [16], [17], [21]. Most of these works consider the case $d = k + r - 1$ to maximally reduce the repair bandwidth, since $\gamma = \frac{d}{(d-k+1)}N$ is a decreasing function of d . This is also the case considered in this work.

Practical systems usually require coding rate in MSR codes k/n to be greater than $1/2$, however in this regime, only four classes of such codes have been reported that can repair both systematic nodes and parity nodes: the $(k+2, k)$ Hadamard design code [9], the $(k+2, k)$ Zigzag code [17] with the repair strategy given in [7], the $(k+r, k)$ modified Zigzag code [21] and the recently constructed $(k+r, k)$ sub-packetization code in [15] with $r \geq 2$. Besides, there are also quite a few constructions of MDS storage codes that can optimally repair only systematic nodes², for examples, the $(k+r, k)$ Hadamard design code [9], [19], the $(k+r, k)$ Zigzag code [17] where $r > 2$, the MDS storage codes based on invariant subspace [6], and the constructions in [20].

Another thread of efforts relevant to our work is the piggybacking design framework [11], [12], which was proposed to reduce the repair bandwidth or reduce the repair-locality of a based MDS code. Specifically, a piggybacking design was presented in [11] which can reduce the repair bandwidth of the parity nodes of the MDS storage codes with the optimal repair property for systematic nodes, however the repair bandwidth of the parity nodes is not optimal. Later on, inspired by the piggybacking method, Yang *et al.* [23] presented a new systematic piggybacking design for MDS storage codes with the optimal repair property for systematic nodes, which can further reduce the bandwidth of the parity nodes, but still suffers a slight loss of optimality.

The scarcity of the constructions for high-rate MSR codes, in contrast to the relative abundance of high-rate MDS storage codes that can optimally repair only systematic nodes, suggests the following conjecture: there might be a fundamental difference between the optimal repair mechanisms of systematic nodes and parity nodes of high-rate MDS storage codes, and constructing high-rate MSR codes (i.e., MDS codes that can optimally repair all nodes) may require more sophisticated techniques. The existing works in [9], [17], [21], [15], [11], [12], [23] also appear to reinforce the belief of such a technical barrier.

In this work, we show that the aforementioned barrier can be completely circumvented. We propose a generic transformation which can convert any non-binary linear MDS storage codes with the optimal repair property only for systematic nodes into new MSR codes that can optimally repair all nodes. This transformation does not increase the alphabet size of the original codes, and only increases the sub-packetization by a factor r . Furthermore, the resultant MSR codes also have the optimal access property for all nodes if the original MDS storage codes have the optimal access property for systematic nodes. It should be noted that our transformation can also be applied to other MDS storage codes without optimal repair property for systematic nodes, and the resultant codes can have the optimal access property for parity nodes.

During this paper was under review, independent of and parallel to our work, Ye and Barg [24], [25] considered the construction of explicit MSR codes that can optimally repair all nodes. Unlike the aforementioned MDS storage codes of systematic form based on generator matrices, the new constructions are given in terms of parity matrices, which does not distinguish between systematic nodes and parity nodes at all. This new development thus essentially provides an alternative solution to overcome the aforementioned barrier between the optimal repair of systematic nodes and the optimal repair of all nodes.

A comparison between the piggyback codes in [11], [23] and the resultant MSR codes obtained by our transformation is provided in Table I, and a comparison between the MSR codes proposed by Ye and Barg and the MSR codes obtained by our transformation is provided in Table II. It is seen from these comparisons that the new transformation has two main advantages: 1) optimal repair property for parity nodes, whereas the repair bandwidth for the parity nodes of the piggyback code in [11] (resp. in [23]) is far from optimality (resp. nearly optimal); 2) lower sub-packetization level and smaller field size in some cases compared to the MSR codes in [24], [25].

²Strictly speaking, the parity nodes can also be repaired, however, not with an optimal repair bandwidth.

TABLE I

A COMPARISON BETWEEN THE PIGGYBACK CODES IN [11], [23] AND THE RESULTANT MSR CODES OBTAINED BY OUR TRANSFORMATION

	Node storage capacity N	Field size	The ratio of repair bandwidth for parity nodes to the optimal value	Remark
Base code	N'	q	$\frac{kr}{k+r-1}$	not optimal
Piggyback code in [11]	$2N'$	q	$> \frac{1}{2} \frac{kr}{k+r-1}$	not optimal
Piggyback code in [23]	rN'	q	$\frac{k+r-2}{k+r-1}$	nearly optimal
Codes obtained by our transformation	rN'	$q(q \geq 3)$	1	optimal

TABLE II

A COMPARISON OF SOME PARAMETERS BETWEEN THE MSR CODES IN [24], [25] AND THE MSR CODES OBTAINED BY APPLYING OUR TRANSFORMATION TO THE HADAMARD DESIGN CODE [9], THE ZIGZAG CODE [17], AND THE OPTIMAL ACCESS CODE TOGETHER WITH THE LONG MDS CODE [20]

	Node storage capacity N	Field size q
Ye-Barg code 1 [24]	r^{k+r}	$q > r(k+r)$
Hadamard design code [9] employing our transformation	r^{k+1}	$q > rk$ [8]
Ye-Barg code 2 [24]	r^{k+r-1}	$q > k+r$
Zigzag code [17] employing our transformation	r^k	$q = 3$ if $r = 2$ $q = 4$ if $r = 3$ $q > r^k \sum_{t=1}^r \binom{k-1}{t-1} \binom{r-1}{t-1}$ if $r > 3$ [20]
Ye-Barg code 3 [25]	$r^{\frac{k}{r}+1}$	$q > k+r$
Optimal access code [20] employing our transformation	$r^{\frac{k}{r}+1}$	$q > k$ [8]
Long MDS code [20] employing our transformation	$r^{\frac{k}{r+1}+1}$	$q > r^{\frac{k}{r+1}+1} \sum_{t=1}^r \binom{k-1}{t-1} \binom{r-1}{t-1}$ [20]

The remainder of this paper is organized as follows. Section II presents some necessary preliminaries. Section III proposes the generic construction which can transform any non-binary MDS storage codes that can optimally repair only systematic nodes to new MSR codes, and then discusses the repair strategy for each node of the new codes and the MDS property. Section IV gives the matrix representation of the new MSR codes. Finally, Section V provides some concluding remarks.

II. PRELIMINARIES

For any two integers $i < j$, denote by $[i, j] = \{i, i+1, \dots, j\}$ and $[i, j) = \{i, i+1, \dots, j-1\}$. Let q be a prime power and \mathbb{F}_q be the finite field with q elements. Assume that a source file comprising of $M = kN$ symbols over a finite field \mathbb{F}_q is equally partitioned into k parts, denoted by $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{k-1}$, respectively, where \mathbf{f}_i is a column vector of length N . The file is encoded by a $(k+r, k)$ storage code and then dispersed across k systematic and r parity storage nodes, each having storage capacity N . More precisely, the first k nodes are systematic nodes, which store the file parts $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{k-1}$ in an uncoded form respectively, and the r parity nodes store linear combinations of $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{k-1}$. In particular, for $i \in [0, r)$, parity node i (node $k+i$) stores $\mathbf{f}_{k+i} = A_{i,0}\mathbf{f}_0 + \dots + A_{i,k-1}\mathbf{f}_{k-1}$, where $A_{i,j}$ is an $N \times N$ matrix over \mathbb{F}_q , termed the *coding matrix* of systematic node j for parity node i , where $j \in [0, k)$. Clearly, the coding matrices composed the generator matrix of a storage code. Table III illustrates the structure of a $(k+r, k)$ storage code. Note that without loss of generality, any vector linear storage code can be written as the form in Table III.

A $(k+r, k)$ MDS code with optimal repair property (i.e., an MSR code) must have the following two properties: (i) *the MDS property*, i.e., the source file can be reconstructed by connecting any k out of the $k+r$ nodes, and (ii) *the optimal repair property*, i.e., any failed node i can be repaired by downloading N/r symbols from each surviving node j , $j \in [0, k+r) \setminus \{i\}$, which can be accomplished by multiplying its original data \mathbf{f}_j with an $N/r \times N$ matrix $S_{i,j}$. Especially, in the literature the matrices $S_{i,j}$'s are called *repair matrices* of node i .

TABLE III
STRUCTURE OF A $(k+r, k)$ STORAGE CODE

	Stored data
Systematic node 0	\mathbf{f}_0
\vdots	\vdots
Systematic node $k-1$	\mathbf{f}_{k-1}
Parity node 0	$\mathbf{f}_k = A_{0,0}\mathbf{f}_0 + \cdots + A_{0,k-1}\mathbf{f}_{k-1}$
Parity node 1	$\mathbf{f}_{k+1} = A_{1,0}\mathbf{f}_0 + \cdots + A_{1,k-1}\mathbf{f}_{k-1}$
\vdots	\vdots
Parity node $r-1$	$\mathbf{f}_{k+r-1} = A_{r-1,0}\mathbf{f}_0 + \cdots + A_{r-1,k-1}\mathbf{f}_{k-1}$

In addition to the optimal repair property, it is also desirable if the nodes have the *optimal access* property. That is, when repairing a failed node, only N/r symbols are accessed at each surviving node, i.e., the minimal amount of data is accessed at each surviving node [18]. That is, to repair node i , all the matrices $S_{i,j}$, $j \in [0, k+r) \setminus \{i\}$ that are used to determine the downloaded information from surviving nodes should have exactly N/r nonzero columns. This appealing property enhances the repair bandwidth requirement, and codes with this property are capable of substantially reducing the disk I/O overhead during the repair process.

III. A GENERIC TRANSFORMATION FOR MSR CODES

In this section, we propose a generic method that can transform any known non-binary linear MDS storage codes with the optimal repair property only for the systematic nodes to new $(k+r, k)$ MSR codes. Before presenting this transformation, an example is provided to illustrate the key idea behind this construction.

A. An example of a new $(9, 6)$ MSR code

Let $\mathbf{f}_0^{(l)}, \dots, \mathbf{f}_5^{(l)}$ and $\mathbf{g}_0^{(l)}, \mathbf{g}_1^{(l)}, \mathbf{g}_2^{(l)}$ be an instance of a known non-binary $(9, 6)$ MDS storage code \mathcal{C} over \mathbb{F}_q , with the optimal repair property holding only for the systematic nodes, where q is odd (for the general construction, q can be even). Three instances of this code, $l = 0, 1, 2$, are used in our construction in this example, and the new $(9, 6)$ MSR code is given in Table IV.

TABLE IV
A NEW $(9, 6)$ MSR CODE

	Instance 0	Instance 1	Instance 2
Systematic node 0 (\mathbf{f}_0)	$\mathbf{f}_0^{(0)}$	$\mathbf{f}_0^{(1)}$	$\mathbf{f}_0^{(2)}$
\vdots	\vdots	\vdots	\vdots
Systematic node 5 (\mathbf{f}_5)	$\mathbf{f}_5^{(0)}$	$\mathbf{f}_5^{(1)}$	$\mathbf{f}_5^{(2)}$
Parity node 0 (\mathbf{f}_6)	$\mathbf{g}_0^{(0)}$	$-\mathbf{g}_1^{(1)} + \mathbf{g}_1^{(0)}$	$-\mathbf{g}_2^{(2)} + \mathbf{g}_2^{(0)}$
Parity node 1 (\mathbf{f}_7)	$\mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)}$	$\mathbf{g}_2^{(1)}$	$-\mathbf{g}_0^{(2)} + \mathbf{g}_0^{(1)}$
Parity node 2 (\mathbf{f}_8)	$\mathbf{g}_2^{(0)} + \mathbf{g}_2^{(2)}$	$\mathbf{g}_0^{(1)} + \mathbf{g}_0^{(2)}$	$\mathbf{g}_1^{(2)}$

Optimal repair of systematic nodes: Let us focus on repairing systematic node 0 of the constructed $(9, 6)$ MSR code, which we claim can be accomplished by downloading the data in Table V. To see this, consider the repair of $\mathbf{f}_0^{(0)}$, i.e., the systematic data from instance 0 stored at node 0, for which the original MDS storage code needs to download

$$S_{0,1}\mathbf{f}_1^{(0)}, S_{0,2}\mathbf{f}_2^{(0)}, \dots, S_{0,5}\mathbf{f}_5^{(0)}, S_{0,6}\mathbf{g}_0^{(0)}, S_{0,7}\mathbf{g}_1^{(0)}, S_{0,8}\mathbf{g}_2^{(0)} \quad (1)$$

for the repair. Comparing these with the downloaded data in the first column of Table V, we see that $S_{0,7}\mathbf{g}_1^{(0)}, S_{0,8}\mathbf{g}_2^{(0)}$ are not directly available. However, $S_{0,7}(\mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)})$, downloaded from parity node 1, and $S_{0,7}(-\mathbf{g}_1^{(1)} + \mathbf{g}_1^{(0)})$, downloaded from

parity node 0, can be utilized to recover $S_{0,7}\mathbf{g}_1^{(0)}$; the data $S_{0,8}\mathbf{g}_2^{(0)}$ can be recovered similarly. At this point, with all the data listed in (1) available, the repair mechanism in the original MDS code \mathcal{C} can be invoked to compute $\mathbf{f}_0^{(0)}$. The repair of $\mathbf{f}_0^{(1)}$ and $\mathbf{f}_0^{(2)}$ can be done in a similar manner, and thus systematic node 0 can indeed be repaired optimally.

TABLE V
DATA DOWNLOADED FROM SURVIVING NODES WHEN REPAIRING SYSTEMATIC NODE 0 OF THE MSR CODE IN TABLE IV

Systematic node 1 (\mathbf{f}_1)	$S_{0,1}\mathbf{f}_1^{(0)}$	$S_{0,1}\mathbf{f}_1^{(1)}$	$S_{0,1}\mathbf{f}_1^{(2)}$
\vdots	\vdots	\vdots	\vdots
Systematic node 5 (\mathbf{f}_5)	$S_{0,5}\mathbf{f}_5^{(0)}$	$S_{0,5}\mathbf{f}_5^{(1)}$	$S_{0,5}\mathbf{f}_5^{(2)}$
Parity node 0 (\mathbf{f}_6)	$S_{0,6}\mathbf{g}_0^{(0)}$	$S_{0,7}(-\mathbf{g}_1^{(1)} + \mathbf{g}_1^{(0)})$	$S_{0,8}(-\mathbf{g}_2^{(2)} + \mathbf{g}_2^{(0)})$
Parity node 1 (\mathbf{f}_7)	$S_{0,7}(\mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)})$	$S_{0,8}\mathbf{g}_2^{(1)}$	$S_{0,6}(-\mathbf{g}_0^{(2)} + \mathbf{g}_0^{(1)})$
Parity node 2 (\mathbf{f}_8)	$S_{0,8}(\mathbf{g}_2^{(0)} + \mathbf{g}_2^{(2)})$	$S_{0,6}(\mathbf{g}_0^{(1)} + \mathbf{g}_0^{(2)})$	$S_{0,7}\mathbf{g}_1^{(2)}$

Optimal repair of parity nodes: Let us focus on the repair of parity node 0, for which the following data are downloaded

$$\mathbf{f}_0^{(0)}, \mathbf{f}_1^{(0)}, \dots, \mathbf{f}_5^{(0)}, \mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)}, \mathbf{g}_2^{(0)} + \mathbf{g}_2^{(2)},$$

i.e., the data in the first column of Table IV. Clearly, $\mathbf{g}_0^{(0)}$ can be computed using $\mathbf{f}_0^{(0)}, \mathbf{f}_1^{(0)}, \dots, \mathbf{f}_5^{(0)}$. To compute $-\mathbf{g}_1^{(1)} + \mathbf{g}_1^{(0)}$ that was stored at parity node 0, observe firstly that $\mathbf{g}_1^{(0)}$ can also be computed using $\mathbf{f}_0^{(0)}, \mathbf{f}_1^{(0)}, \dots, \mathbf{f}_5^{(0)}$, however, this implies that from the downloaded data $\mathbf{g}_1^{(0)} + \mathbf{g}_1^{(1)}$, we can also recover $\mathbf{g}_1^{(1)}$, and subsequently obtain $-\mathbf{g}_1^{(1)} + \mathbf{g}_1^{(0)}$. The other piece of coded data $-\mathbf{g}_2^{(2)} + \mathbf{g}_2^{(0)}$ stored at parity node 0 can be computed similarly. Thus the parity node 0 can indeed be repaired optimally.

Reconstruction: Let us focus on reconstructing the original file by using data stored at nodes 2 to 7; other cases can be addressed similarly. In Table IV, from the symbols that are underlined, we can recover $\mathbf{g}_1^{(0)}$ and $\mathbf{g}_1^{(1)}$. Together with the other data in the first two columns of rows 2 to 7, we now have

$$(\mathbf{f}_2^{(0)}, \dots, \mathbf{f}_5^{(0)}, \mathbf{g}_0^{(0)}, \mathbf{g}_1^{(0)}),$$

$$(\mathbf{f}_2^{(1)}, \dots, \mathbf{f}_5^{(1)}, \mathbf{g}_1^{(1)}, \mathbf{g}_2^{(1)}),$$

from which $(\mathbf{f}_0^{(0)}, \dots, \mathbf{f}_5^{(0)})$ and $(\mathbf{f}_0^{(1)}, \dots, \mathbf{f}_5^{(1)})$ can be reconstructed, respectively, since the base code \mathcal{C} is an MDS code. With these available, $\mathbf{g}_2^{(0)}$ and $\mathbf{g}_0^{(1)}$ can now be computed, and subtracted from the items marked with dashed underline to obtain $\mathbf{g}_2^{(2)}$ and $\mathbf{g}_0^{(2)}$. Together with the other data in the last column of rows 2 to 7, we now also have

$$(\mathbf{f}_2^{(2)}, \dots, \mathbf{f}_5^{(2)}, \mathbf{g}_0^{(2)}, \mathbf{g}_2^{(2)}),$$

from which we can reconstruct $(\mathbf{f}_0^{(2)}, \dots, \mathbf{f}_5^{(2)})$. Thus the original file can indeed be reconstructed using data at nodes 2 to 7.

B. The generic transformation

In this subsection, we present the generic transformation, which utilizes a known linear non-binary $(k+r, k)$ MDS storage code \mathcal{C}_1 with node capacity N , for which the optimal repair property holds only for the systematic nodes. Without loss of generality, it can be assumed that such a base MDS code possesses the following two properties:

- P1: For $j \in [0, r)$, $\mathbf{g}'_j = A_{j,0}\mathbf{f}'_0 + A_{j,1}\mathbf{f}'_1 + \dots + A_{j,k-1}\mathbf{f}'_{k-1}$ is the data stored at parity node j where \mathbf{f}'_i denotes the systematic data stored in systematic node i for $i \in [0, k)$;
- P2: For $i \in [0, k)$, systematic node i can be optimally repaired by downloading $S_{i,j}\mathbf{f}'_j$, $j \in [0, k) \setminus \{i\}$ and $S_{i,k+l}\mathbf{g}'_l$, $l \in [0, r)$.

We next construct a new $(k+r, k)$ MSR code through the following three steps.

Step 1: An intermediate code \mathcal{C}_2 by space sharing r instances of the code \mathcal{C}_1

We construct an intermediate code \mathcal{C}_2 with node capacity rN by space sharing r instances of the code \mathcal{C}_1 . Let $\mathbf{f}_i^{(l)}$ be the data stored at systematic node i of instance l , and $\mathbf{g}_j^{(l)}$ be the data stored in parity node j of instance l where $i \in [0, k)$ and $l, j \in [0, r)$. The new $(k+r, k)$ storage code \mathcal{C}_2 is depicted in Table VI.

TABLE VI
THE NEW CODE \mathcal{C}_2

	Instance 0	Instance 1	...	Instance $r-1$
Systematic node 0	$\mathbf{f}_0^{(0)}$	$\mathbf{f}_0^{(1)}$...	$\mathbf{f}_0^{(r-1)}$
\vdots	\vdots	\vdots	\ddots	\vdots
Systematic node $k-1$	$\mathbf{f}_{k-1}^{(0)}$	$\mathbf{f}_{k-1}^{(1)}$...	$\mathbf{f}_{k-1}^{(r-1)}$
Parity node 0	$\mathbf{g}_0^{(0)}$	$\mathbf{g}_0^{(1)}$...	$\mathbf{g}_0^{(r-1)}$
\vdots	\vdots	\vdots	\ddots	\vdots
Parity node $r-1$	$\mathbf{g}_{r-1}^{(0)}$	$\mathbf{g}_{r-1}^{(1)}$...	$\mathbf{g}_{r-1}^{(r-1)}$

Step 2: An intermediate code \mathcal{C}_3 by permuting the parity data of \mathcal{C}_2

From \mathcal{C}_2 , we construct another intermediate code \mathcal{C}_3 by permuting the parity data while keeping the systematic information intact. Let \mathbf{h}_j denote the data stored in parity node j of the code \mathcal{C}_3 . For convenience, we write \mathbf{h}_j as

$$\mathbf{h}_j = \begin{pmatrix} \mathbf{h}_j^{(0)} \\ \vdots \\ \mathbf{h}_j^{(r-1)} \end{pmatrix}, j \in [0, r)$$

where $\mathbf{h}_j^{(l)}$ ($l \in [0, r)$) is a column vector of length N . Let p_0, p_1, \dots, p_{r-1} be r permutations on $[0, r-1]$. Then $\mathbf{h}_j^{(l)}$ in \mathcal{C}_3 is defined as

$$\mathbf{h}_j^{(l)} = \mathbf{g}_{p_l(j)}^{(l)}, \quad j, l \in [0, r). \quad (2)$$

The parity nodes of the new code \mathcal{C}_3 are depicted in Table VII.

TABLE VII
THE PARITY NODES OF THE CODE \mathcal{C}_3

	Instance 0	Instance 1	...	Instance $r-1$
Parity node 0 (\mathbf{h}_0)	$\mathbf{g}_{p_0(0)}^{(0)}$	$\mathbf{g}_{p_1(0)}^{(1)}$...	$\mathbf{g}_{p_{r-1}(0)}^{(r-1)}$
Parity node 1 (\mathbf{h}_1)	$\mathbf{g}_{p_0(1)}^{(0)}$	$\mathbf{g}_{p_1(1)}^{(1)}$...	$\mathbf{g}_{p_{r-1}(1)}^{(r-1)}$
\vdots	\vdots	\vdots	\ddots	\vdots
Parity node $r-1$ (\mathbf{h}_{r-1})	$\mathbf{g}_{p_0(r-1)}^{(0)}$	$\mathbf{g}_{p_1(r-1)}^{(1)}$...	$\mathbf{g}_{p_{r-1}(r-1)}^{(r-1)}$

Step 3: The MSR code \mathcal{C}_4 by pairing the parity data of \mathcal{C}_3

From the code \mathcal{C}_3 , we construct an MSR code \mathcal{C}_4 by modifying only the parity data while keeping the systematic data intact. Let \mathbf{f}_{k+j} denote the data stored in parity node j of the code \mathcal{C}_4 . For convenience, we write \mathbf{f}_{k+j} as

$$\mathbf{f}_{k+j} = \begin{pmatrix} \mathbf{f}_{k+j}^{(0)} \\ \vdots \\ \mathbf{f}_{k+j}^{(r-1)} \end{pmatrix}, j \in [0, r)$$

where $\mathbf{f}_{k+j}^{(l)}$ ($l \in [0, r)$) is a column vector of length N that defined by

$$\mathbf{f}_{k+j}^{(l)} = \begin{cases} \mathbf{h}_j^{(j)}, & \text{if } j = l \\ \theta_{j,l} \mathbf{h}_j^{(l)} + \mathbf{h}_l^{(j)}, & \text{otherwise} \end{cases} \quad (3)$$

where

$$\{\theta_{j,l}, \theta_{l,j}\} = \{1, a\}, \quad j, l \in [0, r) \text{ with } j \neq l \quad (4)$$

TABLE VIII
THE PARITY NODES OF THE MSR CODE \mathcal{C}_4

Parity node 0 (\mathbf{f}_k)	$\mathbf{h}_0^{(0)}$	$\theta_{0,1}\mathbf{h}_0^{(1)} + \mathbf{h}_1^{(0)}$	\cdots	$\theta_{0,r-1}\mathbf{h}_0^{(r-1)} + \mathbf{h}_{r-1}^{(0)}$
Parity node 1 (\mathbf{f}_{k+1})	$\theta_{1,0}\mathbf{h}_1^{(0)} + \mathbf{h}_0^{(1)}$	$\mathbf{h}_1^{(1)}$	\cdots	$\theta_{1,r-1}\mathbf{h}_1^{(r-1)} + \mathbf{h}_{r-1}^{(1)}$
\vdots	\vdots	\vdots	\ddots	\vdots
Parity node $r-1$ (\mathbf{f}_{k+r-1})	$\theta_{r-1,0}\mathbf{h}_{r-1}^{(0)} + \mathbf{h}_0^{(r-1)}$	$\theta_{r-1,1}\mathbf{h}_{r-1}^{(1)} + \mathbf{h}_1^{(r-1)}$	\cdots	$\mathbf{h}_{r-1}^{(r-1)}$

and $a \in \mathbb{F}_q \setminus \{0, 1\}$. The parity nodes of the new $(k+r, k)$ MSR code \mathcal{C}_4 are depicted in Table VIII.

In the construction above, the repair matrices $S_{i,j}$'s are generic, however in all the aforementioned $(k+r, k)$ MDS storage codes [6], [9], [15], [20], [21] except the Zigzag code [17], simple repair matrices with the form $S_{i,j} = S_i$ are used. In fact, it was shown in [18] that any linear MDS storage code that can optimally repair systematic nodes can be transformed to another linear MDS storage code with such simple repair matrices, however at a cost of sacrificing a systematic node. The proposed code construction is valid for both cases, but the repair strategies for the systematic nodes exhibit different flexibilities depending on whether the condition $S_{i,j} = S_i$ holds.

Theorem 1. *The $(k+r, k)$ MSR code \mathcal{C}_4 has the optimal repair property for the systematic nodes when the permutations satisfy one of the following conditions:*

- (i) p_l ($l \in [0, r)$) can be any permutations if there exist some matrices S_i such that $S_{i,j} = S_i, j \in [0, k+r) \setminus \{i\}$;
- (ii) $p_i(j) = p_j(i)$ for $i, j \in [0, r)$ otherwise.

Proof:

- (i) For $i \in [0, k)$, we prove that systematic node i can be repaired by downloading $S_i \mathbf{f}_j^{(l)}, j \in [0, k+r) \setminus \{i\}, l \in [0, r)$. It suffices to prove that for any $l \in [0, r)$, the data $\mathbf{f}_i^{(l)}$ stored at systematic node i can be repaired with the downloaded data. For any $l \in [0, r) \setminus \{j\}$, it follows from (3) that

$$\begin{cases} S_i \mathbf{f}_{k+j}^{(l)} = \theta_{j,l} S_i \mathbf{h}_j^{(l)} + S_i \mathbf{h}_l^{(j)} \\ S_i \mathbf{f}_{k+l}^{(j)} = \theta_{l,j} S_i \mathbf{h}_l^{(j)} + S_i \mathbf{h}_j^{(l)} \end{cases}.$$

We can now find $S_i \mathbf{h}_j^{(j)}$ and $S_i \mathbf{h}_l^{(j)}$ from $S_i \mathbf{f}_{k+j}^{(l)}$ and $S_i \mathbf{f}_{k+l}^{(j)}$, since $\theta_{j,l} \theta_{l,j} = a \neq 1$ as in (4). Noting that $S_i \mathbf{f}_{k+l}^{(l)} = S_i \mathbf{h}_l^{(l)}$, we have thus collected for any $l \in [0, r)$ the following data

$$\begin{cases} S_i \mathbf{f}_j^{(l)}, & j \in [0, k) \setminus \{i\} \\ S_i \mathbf{h}_s^{(l)}, & s \in [0, r) \end{cases}.$$

Recall from (2) that for each $l \in [0, r)$, $\{\mathbf{h}_0^{(l)}, \dots, \mathbf{h}_{r-1}^{(l)}\}$ is a permutation of $\{\mathbf{g}_0^{(l)}, \dots, \mathbf{g}_{r-1}^{(l)}\}$. It follows that indeed we already have all the data $S_i \mathbf{f}_j^{(l)}$ ($j \in [0, k) \setminus \{i\}$) and $S_i \mathbf{g}_s^{(l)}$ ($s \in [0, r)$), and thus using the repair mechanism of the base MDS code \mathcal{C}_1 , the data $\mathbf{f}_i^{(l)}$ can be recovered.

- (ii) For $i \in [0, k)$, we prove that systematic node i can be repaired by downloading $S_{i,s} \mathbf{f}_s^{(l)}$ and $S_{i,k+p_l(j)} \mathbf{f}_{k+j}^{(l)}$ with $s \in [0, k) \setminus \{i\}$ and $j, l \in [0, r)$. For this purpose, it suffices to prove that for any $l \in [0, r)$, we can obtain $S_{i,k+j} \mathbf{g}_j^{(l)}$ with $j \in [0, r)$ from the downloaded data, and then the repair mechanism of the base MDS code \mathcal{C}_1 can be invoked to complete the repair.

For $j \in [0, r) \setminus \{l\}$, from (3), we have

$$\begin{cases} S_{i,k+p_l(j)} \mathbf{f}_{k+j}^{(l)} = \theta_{j,l} S_{i,k+p_l(j)} \mathbf{h}_j^{(l)} + S_{i,k+p_l(j)} \mathbf{h}_l^{(j)} \\ S_{i,k+p_j(l)} \mathbf{f}_{k+l}^{(j)} = \theta_{l,j} S_{i,k+p_j(l)} \mathbf{h}_l^{(j)} + S_{i,k+p_j(l)} \mathbf{h}_j^{(l)} \end{cases}.$$

Together with the facts that $p_l(j) = p_j(l)$ and $\theta_{j,l} \theta_{l,j} = a \neq 1$ as in (4), we can find $S_{i,k+p_l(j)} \mathbf{h}_j^{(l)}, j \in [0, r) \setminus \{l\}$ from the above equations. Since $S_{i,k+p_l(l)} \mathbf{h}_l^{(l)} = S_{i,k+p_l(l)} \mathbf{f}_{k+l}^{(l)}$ is also available, we have thus collected all $S_{i,k+p_l(j)} \mathbf{h}_j^{(l)}$ (i.e., $S_{i,k+p_l(j)} \mathbf{g}_{p_l(j)}^{(l)}, j \in [0, r)$), or equivalently, we have collected all $S_{i,k+j} \mathbf{g}_j^{(l)}, j \in [0, r)$ since p_l is a permutation. This completes the proof.

Remark 1. There are many choices of the permutations p_0, \dots, p_{r-1} satisfying the condition $p_i(j) = p_j(i)$ in Theorem 1 item (ii), for example,

$$p_i(j) = (i + j) \bmod r, \quad i, j \in [0, r),$$

as we used in Section III-A.

The following result is a direct consequence of Theorem 1 and the definition of optimal access.

Corollary 1. The $(k + r, k)$ MSR code \mathcal{C}_4 has the optimal access property for systematic nodes if the $(k + r, k)$ MDS storage code \mathcal{C}_1 has the optimal access property for systematic nodes.

Theorem 2. The $(k + r, k)$ MSR code \mathcal{C}_4 has the optimal repair property for the parity nodes.

Proof: We show that for any $j \in [0, r)$, the data in parity node j can be repaired by downloading $\mathbf{f}_{k+l}^{(j)}$, $l \in [0, r) \setminus \{j\}$, and $\mathbf{f}_i^{(j)}$, $i \in [0, k)$.

Firstly, using $\mathbf{f}_i^{(j)}$, $i \in [0, k)$, we can compute $\mathbf{g}_s^{(j)}$, $s \in [0, r)$, and then obtain $\mathbf{h}_s^{(j)}$, $s \in [0, r)$, since $\{\mathbf{h}_0^{(j)}, \dots, \mathbf{h}_{r-1}^{(j)}\}$ is a permutation of $\{\mathbf{g}_0^{(j)}, \dots, \mathbf{g}_{r-1}^{(j)}\}$ according to (2). Next, for any $l \in [0, r) \setminus \{j\}$, from the downloaded data $\mathbf{f}_{k+l}^{(j)} = \theta_{l,j} \mathbf{h}_l^{(j)} + \mathbf{h}_j^{(l)}$, we can obtain $\mathbf{h}_j^{(l)}$ by subtracting $\mathbf{h}_l^{(j)}$ from $\mathbf{f}_{k+l}^{(j)}$, therefore compute $\mathbf{f}_{k+j}^{(l)} = \theta_{j,l} \mathbf{h}_j^{(l)} + \mathbf{h}_l^{(j)}$. Finally, since $\mathbf{f}_{k+j}^{(j)} = \mathbf{h}_j^{(j)}$, which has already been computed in the first step, parity node j can indeed be repaired optimally. ■

The following result is also a direct consequence of Theorem 2 and the definition of optimal access.

Corollary 2. The $(k + r, k)$ MSR code \mathcal{C}_4 has the optimal access property for the parity nodes.

Finally, we need to show that the MDS property holds for the $(k + r, k)$ MSR code \mathcal{C}_4 .

Theorem 3. The code \mathcal{C}_4 has the MDS property.

Proof: The code \mathcal{C}_4 possesses the MDS property if any k out of the $k + r$ nodes can reconstruct the original file, i.e., the systematic data $\mathbf{f}_i^{(l)}$, $i \in [0, k)$ and $l \in [0, r)$. We discuss the reconstruction in two cases.

- (i) When connecting to all k systematic nodes: there is nothing to prove.
- (ii) When connecting to $k - t$ systematic nodes and t parity nodes where $1 \leq t \leq \min\{r, k\}$: we assume that $I = \{i_0, i_1, \dots, i_{t-1}\}$ is the set of the systematic nodes which are not connected and $J = \{j_0, j_1, \dots, j_{t-1}\}$ is the set of the parity nodes which are connected, where $0 \leq i_0 < \dots < i_{t-1} < k$ and $0 \leq j_0 < \dots < j_{t-1} < r$. Denote by $\{j_t, \dots, j_{r-1}\} = [0, r) \setminus J$. Then we have the following data in Table IX from the parity nodes that are connected.

TABLE IX

$\mathbf{h}_{j_0}^{(j_0)}$	$\theta_{j_0, j_1} \mathbf{h}_{j_0}^{(j_1)} + \mathbf{h}_{j_1}^{(j_0)}$	\dots	$\theta_{j_0, j_{t-1}} \mathbf{h}_{j_0}^{(j_{t-1})} + \mathbf{h}_{j_{t-1}}^{(j_0)}$	$\theta_{j_0, j_t} \mathbf{h}_{j_0}^{(j_t)} + \mathbf{h}_{j_t}^{(j_0)}$	\dots	$\theta_{j_0, j_{r-1}} \mathbf{h}_{j_0}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_0)}$
$\theta_{j_1, j_0} \mathbf{h}_{j_1}^{(j_0)} + \mathbf{h}_{j_0}^{(j_1)}$	$\mathbf{h}_{j_1}^{(j_1)}$	\dots	$\theta_{j_1, j_{t-1}} \mathbf{h}_{j_1}^{(j_{t-1})} + \mathbf{h}_{j_{t-1}}^{(j_1)}$	$\theta_{j_1, j_t} \mathbf{h}_{j_1}^{(j_t)} + \mathbf{h}_{j_t}^{(j_1)}$	\dots	$\theta_{j_1, j_{r-1}} \mathbf{h}_{j_1}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_1)}$
\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
$\theta_{j_{t-1}, j_0} \mathbf{h}_{j_{t-1}}^{(j_0)} + \mathbf{h}_{j_0}^{(j_{t-1})}$	$\theta_{j_{t-1}, j_1} \mathbf{h}_{j_{t-1}}^{(j_1)} + \mathbf{h}_{j_1}^{(j_{t-1})}$	\dots	$\mathbf{h}_{j_{t-1}}^{(j_{t-1})}$	$\theta_{j_{t-1}, j_t} \mathbf{h}_{j_{t-1}}^{(j_t)} + \mathbf{h}_{j_t}^{(j_{t-1})}$	\dots	$\theta_{j_{t-1}, j_{r-1}} \mathbf{h}_{j_{t-1}}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_{t-1})}$

Note that we can get the data in Table X from that in Table IX since the data in the first t columns of Table X can be clearly computed from the data in the first t columns of Table IX by solving systems of linear independent equations (Specifically for $t = 1$, no equations need to be solved).

For each $s \in J$, combining the parity data in the first t columns of Table X with the systematic data $\mathbf{f}_i^{(s)}$ ($i \in [0, k-1] \setminus I$) in the $k - t$ systematic nodes of code \mathcal{C}_4 that are connected, we can obtain $\mathbf{h}_u^{(s)}$, $u \in [0, r) \setminus \{J\}$ by means of the MDS property of code \mathcal{C}_1 and the fact that $\{\mathbf{h}_0^{(j)}, \dots, \mathbf{h}_{r-1}^{(j)}\}$ is a permutation of $\{\mathbf{g}_0^{(j)}, \dots, \mathbf{g}_{r-1}^{(j)}\}$. Since we can compute the data in Table X that marked with dash underline, we obtain the data in Table XI. That is for any $l \in [0, r)$, the parity

TABLE X

$\mathbf{h}_{j_0}^{(j_0)}$	$\mathbf{h}_{j_0}^{(j_1)}$	\dots	$\mathbf{h}_{j_0}^{(j_{t-1})}$	$\theta_{j_0,j_t} \mathbf{h}_{j_0}^{(j_t)} + \mathbf{h}_{j_t}^{(j_0)}$	\dots	$\theta_{j_0,j_{r-1}} \mathbf{h}_{j_0}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_0)}$
$\mathbf{h}_{j_1}^{(j_0)}$	$\mathbf{h}_{j_1}^{(j_1)}$	\dots	$\mathbf{h}_{j_1}^{(j_{t-1})}$	$\theta_{j_1,j_t} \mathbf{h}_{j_1}^{(j_t)} + \mathbf{h}_{j_t}^{(j_1)}$	\dots	$\theta_{j_1,j_{r-1}} \mathbf{h}_{j_1}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_1)}$
\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
$\mathbf{h}_{j_{t-1}}^{(j_0)}$	$\mathbf{h}_{j_{t-1}}^{(j_1)}$	\dots	$\mathbf{h}_{j_{t-1}}^{(j_{t-1})}$	$\theta_{j_{t-1},j_t} \mathbf{h}_{j_{t-1}}^{(j_t)} + \mathbf{h}_{j_t}^{(j_{t-1})}$	\dots	$\theta_{j_{t-1},j_{r-1}} \mathbf{h}_{j_{t-1}}^{(j_{r-1})} + \mathbf{h}_{j_{r-1}}^{(j_{t-1})}$

data $\mathbf{h}_s^{(l)}$, i.e., $\mathbf{g}_{p_l(s)}^{(l)}$, $s \in J$ are available, together with $\mathbf{f}_i^{(l)}$, $i \in [0, k-1] \setminus I$ in the $k-t$ systematic nodes which are connected, we can recover the remaining systematic data $\mathbf{f}_{i_0}^{(l)}, \dots, \mathbf{f}_{i_{t-1}}^{(l)}$ by means of the MDS property of code \mathcal{C}_1 .

TABLE XI

$\mathbf{h}_{j_0}^{(j_0)}$	$\mathbf{h}_{j_0}^{(j_1)}$	\dots	$\mathbf{h}_{j_0}^{(j_{t-1})}$	$\mathbf{h}_{j_0}^{(j_t)}$	\dots	$\mathbf{h}_{j_0}^{(j_{r-1})}$
$\mathbf{h}_{j_1}^{(j_0)}$	$\mathbf{h}_{j_1}^{(j_1)}$	\dots	$\mathbf{h}_{j_1}^{(j_{t-1})}$	$\mathbf{h}_{j_1}^{(j_t)}$	\dots	$\mathbf{h}_{j_1}^{(j_{r-1})}$
\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots
$\mathbf{h}_{j_{t-1}}^{(j_0)}$	$\mathbf{h}_{j_{t-1}}^{(j_1)}$	\dots	$\mathbf{h}_{j_{t-1}}^{(j_{t-1})}$	$\mathbf{h}_{j_{t-1}}^{(j_t)}$	\dots	$\mathbf{h}_{j_{t-1}}^{(j_{r-1})}$

■

Remark 2. In fact, from the transformation and the proofs in this section, one can easily get that this transformation can also be applied to MDS storage codes even without the optimal repair property for systematic nodes, the resultant codes maintain the MDS property and have the optimal access property for parity nodes. Furthermore, the resultant MDS codes keep the repair bandwidth ratio (i.e., the repair bandwidth normalized by the file size).

IV. MATRIX INTERPRETATION OF THE NEW CONSTRUCTION

In the literature, MSR codes are often analyzed and their correctness proved from the viewpoint of the coding matrix; see, e.g., [6], [9], [20]. In this section, we take an alternative view and provide the coding matrices and repair matrices of the $(k+r, k)$ MSR code \mathcal{C}_4 .

First of all, we give some introduction about a $(k+r, k)$ storage code from the viewpoint of the coding matrix.

To guarantee the MDS property of a $(k+r, k)$ storage code with coding matrices $A_{i,j}$, $i \in [0, r)$ and $j \in [0, k)$, any $t \times t$ sub-block matrix of

$$\begin{pmatrix} A_{0,0} & A_{0,1} & \dots & A_{0,k-1} \\ A_{1,0} & A_{1,1} & \dots & A_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r-1,0} & A_{r-1,1} & \dots & A_{r-1,k-1} \end{pmatrix}$$

needs to be nonsingular, where $1 \leq t \leq \min\{r, k\}$, from the viewpoint of the coding matrix [20].

When repairing systematic node i where $i \in [0, k)$, according to the data downloaded from the r parity nodes and the structure of a $(k+r, k)$ storage code in Table III, we obtain the following system of linear equations

$$\begin{pmatrix} S_{i,k} \mathbf{f}_k \\ S_{i,k+1} \mathbf{f}_{k+1} \\ \vdots \\ S_{i,k+r-1} \mathbf{f}_{k+r-1} \end{pmatrix} = \underbrace{\begin{pmatrix} S_{i,k} A_{0,i} \\ S_{i,k+1} A_{1,i} \\ \vdots \\ S_{i,k+r-1} A_{r-1,i} \end{pmatrix}}_{\text{useful data}} \mathbf{f}_i + \sum_{j=0, j \neq i}^{k-1} \underbrace{\begin{pmatrix} S_{i,k} A_{0,j} \\ S_{i,k+1} A_{1,j} \\ \vdots \\ S_{i,k+r-1} A_{r-1,j} \end{pmatrix}}_{\text{interference by } \mathbf{f}_j} \mathbf{f}_j.$$

Then, the optimal repair property stipulates that the coefficient matrix of the useful data in the above system of linear equations is of full rank, and the interference terms caused by \mathbf{f}_j can be cancelled by $S_{i,j} \mathbf{f}_j$ downloaded from systematic node j for all

$j \in [0, k) \setminus \{i\}$, i.e.,

$$\text{rank} \begin{pmatrix} S_{i,k} A_{0,i} \\ S_{i,k+1} A_{1,i} \\ \vdots \\ S_{i,k+r-1} A_{r-1,i} \end{pmatrix} = N \quad \text{for } i \in [0, k), \quad (5)$$

and

$$\text{rank} \begin{pmatrix} S_{i,j} \\ S_{i,k} A_{0,j} \\ S_{i,k+1} A_{1,j} \\ \vdots \\ S_{i,k+r-1} A_{r-1,j} \end{pmatrix} = \frac{N}{r} \quad \text{for } i, j \in [0, k) \text{ with } i \neq j. \quad (6)$$

Next, we give the coding matrices of the $(k+r, k)$ MSR code \mathcal{C}_4 .

Proposition 1. Let \mathcal{C}_1 be a known $(k+r, k)$ MDS storage code over \mathbb{F}_q ($q \geq 3$) with node capacity N and the optimal repair property for systematic nodes, and let $A_{i,j}$, $i \in [0, r)$ and $j \in [0, k)$, be the $N \times N$ coding matrices of \mathcal{C}_1 . Denote by $B_{i,j}$, $i \in [0, r)$ and $j \in [0, k)$, the $rN \times rN$ coding matrices of the code \mathcal{C}_4 obtained in Section IV with \mathcal{C}_1 being the base code, i.e., parity node i of \mathcal{C}_4 stores

$$\mathbf{f}_{k+i} = \sum_{j=0}^{k-1} B_{i,j} \mathbf{f}_j, \quad i \in [0, r).$$

Then, the (l, s) th block of $B_{i,j}$ can be written as

$$B_{i,j}(l, s) = \begin{cases} A_{p_i(i),j}, & \text{if } l = s = i \\ A_{p_i(l),j}, & \text{if } s = i \neq l \\ \theta_{i,l} A_{p_l(i),j}, & \text{if } l = s \neq i \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

where $l, s \in [0, r)$, i.e.,

$$B_{i,j} = \begin{pmatrix} A_{p_i(i),j} & & & & & \\ A_{p_i(i+1),j} & \theta_{i,i+1} A_{p_{i+1}(i),j} & & & & \\ A_{p_i(i+2),j} & & \theta_{i,i+2} A_{p_{i+2}(i),j} & & & \\ \vdots & & & \ddots & & \\ A_{p_i(i+r-2),j} & & & & \theta_{i,i+r-2} A_{p_{i+r-2}(i),j} & \\ A_{p_i(i+r-1),j} & & & & & \theta_{i,i+r-1} A_{p_{i+r-1}(i),j} \end{pmatrix} \begin{matrix} \xrightarrow{\text{cyclically shift } i \text{ blocks}} \\ \downarrow \text{cyclically shift } i \text{ blocks} \end{matrix}$$

where the subscripts i, j, l, s in $p_{i,j}$ and $\theta_{i,s}$ are computed modulo r .

Proof: Since the systematic data \mathbf{f}_j ($j \in [0, k)$) consists of r instances, for ease of analysis, we take $B_{i,j}$ as an $r \times r$ block matrix with $B_{i,j}[l]$ being the block row l of the matrix $B_{i,j}$ and $B_{i,j}(l, s)$ being the (l, s) th block of $B_{i,j}$, which are respectively an $N \times rN$ matrix and an $N \times N$ matrix.

For any $i, l \in [0, r)$ with $i \neq l$, by (2), (3) and (7), we have

$$\mathbf{f}_{k+i}^{(l)} = \theta_{i,l} \mathbf{h}_i^{(l)} + \mathbf{h}_l^{(i)} = \theta_{i,l} \mathbf{g}_{p_l(i)}^{(l)} + \mathbf{g}_{p_i(l)}^{(i)} = \theta_{i,l} \sum_{j=0}^{k-1} A_{p_l(i),j} \mathbf{f}_j^{(l)} + \sum_{j=0}^{k-1} A_{p_i(l),j} \mathbf{f}_j^{(i)} = \sum_{j=0}^{k-1} B_{i,j}[l] \mathbf{f}_j,$$

and

$$\mathbf{f}_{k+i}^{(i)} = \mathbf{h}_i^{(i)} = \mathbf{g}_{p_i(i)}^{(i)} = \sum_{j=0}^{k-1} A_{p_i(i),j} \mathbf{f}_j^{(i)} = \sum_{j=0}^{k-1} B_{i,j}[i] \mathbf{f}_j,$$

which implies

$$B_{i,j}[l] = \begin{pmatrix} \cdots & \overset{i}{A_{p_i(l),j}} & \cdots & \overset{l}{\theta_{i,l}A_{p_l(i),j}} & \cdots \end{pmatrix}$$

and

$$B_{i,j}[i] = \begin{pmatrix} \cdots & \overset{i}{A_{p_i(i),j}} & \cdots \end{pmatrix},$$

thus we can get the desired result. ■

Finally, we give the repair matrices of the $(k+r, k)$ MSR code \mathcal{C}_4 .

According to the proof of Theorem 1, systematic node i can be optimally repaired by downloading $R_{i,j}\mathbf{f}_j$, $j \in [0, k+r) \setminus \{i\}$ where

$$R_{i,j} = \text{blkdiag}(\underbrace{S_i, \dots, S_i}_r), \quad i \in [0, k), \quad j \in [0, k+r) \setminus \{i\}$$

if for any $i \in [0, k)$, $S_{i,j} = S_i$, $j \in [0, k+r) \setminus \{i\}$ for some matrices S_i ; otherwise,

$$\begin{cases} R_{i,j} = \text{blkdiag}(\underbrace{S_{i,j}, \dots, S_{i,j}}_r), & j \in [0, k) \setminus \{i\} \\ R_{i,k+j} = \text{blkdiag}(\underbrace{S_{i,k+p_0(j)}, \dots, S_{i,k+p_{r-1}(j)}}_r), & j \in [0, r) \end{cases}$$

According to the proof of Theorem 2, parity node i can be optimally repaired by downloading $R_{k+i,j}\mathbf{f}_j$, $j \in [0, k+r) \setminus \{k+i\}$ where

$$R_{k+i,j} = \begin{pmatrix} \cdots & \overset{i}{I_N} & \cdots \end{pmatrix}, \quad i \in [0, r), \quad j \in [0, k+r) \text{ with } j \neq k+i$$

with all the omitted blocks being the zero matrix.

As an alternative method to verify the optimal repair property for systematic nodes of the MSR code \mathcal{C}_4 , one can check that the coding matrices $B_{i,j}$ and repair matrices $R_{j,l}$, $i \in [0, r)$, $j \in [0, k)$, and $l \in [0, k+r)$ with $l \neq j$ satisfy the rank conditions in (5) and (6), that is

$$\text{rank} \begin{pmatrix} R_{i,k}B_{0,i} \\ R_{i,k+1}B_{1,i} \\ \vdots \\ R_{i,k+r-1}B_{r-1,i} \end{pmatrix} = rN \quad \text{for } i \in [0, k),$$

and

$$\text{rank} \begin{pmatrix} R_{i,j} \\ R_{i,k}B_{0,j} \\ R_{i,k+1}B_{1,j} \\ \vdots \\ R_{i,k+r-1}B_{r-1,j} \end{pmatrix} = N \quad \text{for } i, j \in [0, k) \text{ with } i \neq j.$$

In addition, one can also verify the MDS property of the code \mathcal{C}_4 by checking that any $t \times t$ sub-block matrix of

$$B = \begin{pmatrix} B_{0,0} & \cdots & B_{0,k-1} \\ \vdots & \ddots & \vdots \\ B_{r-1,0} & \cdots & B_{r-1,k-1} \end{pmatrix}$$

is nonsingular, where $1 \leq t \leq \min\{r, k\}$.

V. CONCLUDING REMARKS

In this paper, we studied the problem of designing MSR codes by presenting a generic transformation from known non-binary MDS storage codes with the optimal repair property only for systematic nodes to MSR codes. The new MSR codes have the optimal access property for parity nodes, and maintain the optimal access property for systematic nodes if the original $(k+r, k)$ MDS storage codes have the optimal access property for systematic nodes. Furthermore, this generic transformation can also be applied to MDS storage codes even without the optimal repair property for systematic nodes.

Recently, there are several new constructions of MDS codes over large enough finite field [4], [22] that allow the number of helper nodes to be strictly less than $k+r-1$ or multiple failed nodes to be repaired simultaneously, for which the transformation proposed in this work does not apply. Extending the proposed transformation to such cases is part of our ongoing work.

REFERENCES

- [1] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," in *Proc. 1st Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004, pp. 1-14.
- [2] F. Dabek, J. Li, E. Sit, J. Robertson, M. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *Proc. 1st Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004, pp. 1-14.
- [3] A. G. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4539-4551, Sep. 2010.
- [4] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," [Online]. Available at: arXiv: 1602.04496v1 [cs.IT]
- [5] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," in *Proc. 2012 USENIX Annual Technical Conference*, Boston, MA, Jun. 2012, pp. 1-12.
- [6] J. Li, X.H. Tang, and U. Parampalli, "A framework of constructions of minimal storage regenerating codes with the optimal access/update property," *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1920-1932, Apr. 2015.
- [7] J. Li and X.H. Tang, "Optimal exact repair strategy for the parity nodes of the $(k+2, k)$ Zigzag code," *IEEE Trans. Inform. Theory*, vol. 62, no. 9, pp. 4848-4856, Sep. 2016.
- [8] Y. Liu and X.H. Tang, "Determining the field size of several high-rate MDS storage codes," preprint.
- [9] D.S. Papailiopoulos, A.G. Dimakis, and V.R. Cadambe, "Repair optimal erasure codes through hadamard designs," *IEEE Trans. Inform. Theory*, vol. 59, no. 5, pp. 3021-3037, May 2013.
- [10] K.V. Rashmi, N.B. Shah, and P.V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 5227-5239, Aug. 2011.
- [11] K.V. Rashmi, N.B. Shah, and K. Ramchandran, "A piggybacking design framework for read-and download-efficient distributed storage codes," [Online]. Available at: arXiv: 1302.5872 [cs.IT]
- [12] K.V. Rashmi, N.B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran, "A 'hitchhiker's' guide to fast and efficient data reconstruction in erasure-coded data centers," in *Proc. ACM SIGCOMM*, pp. 331C342, 2014.
- [13] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300-304, Jun. 1960.
- [14] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Comput.*, vol. 5, no. 5, pp. 40-49, Sep.-Oct. 2001.
- [15] B. Sasidharan, G.K. Agarwal, and P.V. Kumar, "A high-rate MSR code with polynomial sub-packetization level," in *Proc. IEEE Int. Symp. Inform. Theory*, Hong Kong, China, Jun. 2015, pp. 2051-2055.
- [16] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Trans. Inform. Theory*, vol. 57, no. 3, pp. 1425-1442, Mar. 2011.
- [17] T. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inform. Theory*, vol. 59, no. 3, pp. 1597-1616, Mar. 2013.
- [18] T. Tamo, Z. Wang, and J. Bruck, "Access versus bandwidth in codes for storage," *IEEE Trans. Inform. Theory*, vol. 60, no. 4, pp. 2028-2037, Apr. 2014.
- [19] X.H. Tang, B. Yang, J. Li, and H.D.L. Hollmann, "A new repair strategy for the hadamard minimum storage regenerating codes for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 61, no. 10, pp. 5271-5279, Oct. 2015.
- [20] Z. Wang, T. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Trans. Inform. Theory*, vol. 62, no. 8, pp. 4466-4480, Aug. 2016.
- [21] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 2011, pp. 1374-1381.
- [22] Z. Wang, I. Tamo, and J. Bruck, "Optimal rebuilding of multiple erasures in MDS codes," *IEEE Trans. Inform. Theory*, to appear.
- [23] B. Yang, X. H. Tang, and J. Li, "A systematic piggybacking design for minimum storage regenerating codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 11, pp. 5779-5786, Nov. 2015.
- [24] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," [Online]. Available at: arXiv: 1604.00454v2 [cs.IT]
- [25] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," [Online]. Available at: arXiv: 1605.08630 [cs.IT]